

SPECIFICATION AMENDMENTS

Please replace the paragraph [0027] with the following rewritten paragraph:

[0027] The second step involves using the filter fields from step 1 to perform an LPM lookup on each field in parallel in its respective filter table. Any LPM algorithm could be used to perform these searches, including a straight array lookup or any tree lookup. The choice of lookup algorithm would depend on the number and size of the filter fields. The result of each lookup will be a bit field, 128 bits long in the case of IPv6, in which each bit indicates a prefix length for which there is a prefix that potentially matches the search key. As a result of these lookups, several sets of potential prefix lengths are determined. These sets of prefix lengths can be further reduced by finding the intersection of the sets to produce a definitive set of prefix lengths. As such, the references in the Figures to lengths should be understood to be references to prefix lengths. This final set of potential prefix lengths may contain one or more set bits, which can be further classified. One of the set bits potentially indicates the length of the longest matching prefix, any bits higher than this bit are false positives, and all bits below this bit indicate shorter matching prefixes and additional false positives. It is important to note that the filtering process would never result in false negatives, in which an actual matching prefix does not have its length indicated in the final set of potential prefix lengths.

Please replace the paragraph [0034] with the following rewritten paragraph:

[0034] A further variation involves an ideal offset filter which is described in greater detail in co-pending U.S. application No. 10/718,524, filed November 24, 2003. The contents of the co-pending application are incorporated herein by reference. Figure 4 shows a sliding window of fixed size to select the ideal groupings of bits to best group addresses in order to reduce the number of prefix lengths within a group. In this embodiment the bits from the sliding window are used as the extra filter. There are two further aspects of this scenario. The first is simply to use statistics of the ideal location of the sliding window to build the filter. In this case the position of the sliding window is fixed as shown in Figure 5. The second scenario is to dynamically determine the ideal position of the window based on the current routing table.